UNSUPERVISED DOMAIN ADAPTATION FOR ROBUST SPEECH RECOGNITION VIA VARIATIONAL AUTOENCODER-BASED DATA AUGMENTATION

Wei-Ning Hsu[*]*, Yu Zhang, James Glass*

Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA
{wnhsu,yzhang87,glass}@csail.mit.edu

ABSTRACT

Domain mismatch between training and testing can lead to significant degradation in performance in many machine learning scenarios. Unfortunately, this is not a rare situation for automatic speech recognition deployments in real-world applications. Research on robust speech recognition can be regarded as trying to overcome this domain mismatch issue. In this paper, we address the unsupervised domain adaptation problem for robust speech recognition, where both source and target domain speech are available, but word transcripts are only available for the source domain speech. We present novel augmentation-based methods that transform speech in a way that does not change the transcripts. Specifically, we first train a variational autoencoder on both source and target domain data (without supervision) to learn a latent representation of speech. We then transform nuisance attributes of speech that are irrelevant to recognition by modifying the latent representations, in order to augment labeled training data with additional data whose distribution is more similar to the target domain. The proposed method is evaluated on the CHiME-4 dataset and reduces the absolute word error rate (WER) by as much as 35% compared to the non-adapted baseline.

Index Terms— unsupervised domain adaptation, robust speech recognition, variational autoencoder, data augmentation

1. INTRODUCTION

Recent advances in neural network-based acoustic models [1, 2, 3, 4, 5] have greatly improved the performance of automatic speech recognition (ASR) systems, enabling more applications to adopt speech-based human-machine interaction. With the increasing use of ASR systems in everyday life, ASR robustness under adverse conditions becomes more essential than ever. Some robust ASR research focuses on enhancing speech, by applying beam-forming techniques [6, 7], estimating noise masks [8, 9], or training denoising models [10, 11], etc. Other research extracts robust acoustic features [12, 13, 14, 15] that are intended to be invariant for ASR even in adverse environments. Another line of research investigates modeling techniques, including, but not limited to, model adaptation [16, 17], and training models on data in adverse conditions [18, 19]. Over the past decade, neural network-based acoustic models have come to dominate the ASR field. To utilize the full capacity of neural network-based acoustic models, it is often a good strategy to train a model with as much, and as diverse a dataset as possible [20].

In this paper, we consider a highly adverse scenario, where both source and target domain speech are available, but word transcripts are only available for the source domain data. We present novel augmentation-based methods that transform speech, but, do not require altering existing transcripts. Specifically, we first train an unsupervised sequence-to-sequence recurrent variational autoencoder (VAE) on both source and target domain data to learn a latent representation of speech. We then transform "nuisance" attributes of speech, such as speaker identities and noise types, that do not contain linguistic information, and are thus irrelevant to ASR, by modifying the latent representation, in order to create additional labeled training data whose distribution is more similar to the target domain. We evaluate the proposed methods on data from the CHiME-4 challenge [21], which is highly mismatched from the source domain data, WSJ0 [22]. The proposed method reduces the absolute word error rate (WER) by as much as 35% compared to a non-adapted baseline.

The rest of the paper is organized as follows. In Section 2, we introduce the VAE model, and present the augmentation methods in Section 3. Related work is discussed in Section 4. Experimental setup and results are shown in Section 5 and 6 respectively. Lastly, we conclude our work and discuss about future work in Section 7.

2. VARIATIONAL AUTOENCODER MODEL

2.1. Variational Autoencoder

Consider a speech dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ consisting of N i.i.d. samples of observed variables $\boldsymbol{x}$. We assume that the data are generated from some random process that involves latent variables $\boldsymbol{z}$ as follows: (1) a latent variable $\boldsymbol{z}$ is drawn from a prior distribution $p_{\theta^*}(\boldsymbol{z})$; (2) an observed variable $\boldsymbol{x}$ is drawn from a conditional distribution $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$. We assume that both $p_{\theta^*}(\boldsymbol{z})$ and $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$ come from some distribution family parameterized by θ. To learn this generative process with the presence of only observed data, it is often required to estimate the posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$; however, with moderately complex conditional distributions $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, true posterior distributions are generally intractable.

A variational autoencoder [23] addresses this issue by introducing an encoder and a decoder. The decoder maps the latent variable $\boldsymbol{z}$ to the conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, while the encoder maps the observed variable $\boldsymbol{x}$ to the approximated posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, an approximation of the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. The log marginal likelihood of the dataset is the sum of individual log marginal likelihood $\log p_\theta(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(N)}) = \sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}^{(i)})$, and each can be rewritten as:

$$\begin{aligned}
&\log p_\theta(\boldsymbol{x}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) + \mathcal{L}(\theta, \phi; \boldsymbol{x}) \\
&\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) \\
&= -D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})], \quad (1)
\end{aligned}$$

[*]This research was supported by a TUSA Fellowship and by PingAn.

where $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ is the variational lower bound to the log marginal likelihood of data $\boldsymbol{x}$. Since the exact ML estimation for the parameter θ is intractable, we optimize the variational lower bounds in Eq.1 of the dataset instead to obtain the approximate ML estimation for θ. Under certain mild condition for $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, we can rewrite the second term in Eq.1 to be the expectation taken over an auxiliary noise distribution such that the Monte Carlo estimation of the expectation is differentiable w.r.t. ϕ. By maximizing Eq.1, we can apply stochastic gradient based approaches to jointly optimize θ and ϕ.

2.2. Sequence-to-Sequence Recurrent VAE Architecture

In this work, we use a filter bank for the frame-level representation of speech, which are extracted every 10ms using a 25ms window. We let the observed data $\boldsymbol{x} = \{x_1, \cdots, x_{20}\}$ be a sequence of 20 frames, roughly at the scale of a syllable. VAEs are applied to learn the generative process of syllable-level speech segments. For the model we consider here, both the conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and the approximate posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ are diagonal Gaussian distributions:

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; f_{\boldsymbol{\mu_z}}(\boldsymbol{x}; \theta), \exp(f_{\log \sigma^2_{\boldsymbol{z}}}(\boldsymbol{x}; \theta)))$$
$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; g_{\boldsymbol{\mu_x}}(\boldsymbol{z}; \phi), \exp(g_{\log \sigma^2_{\boldsymbol{x}}}(\boldsymbol{z}; \phi))),$$

of which the mean ($f_{\boldsymbol{\mu_z}}(\boldsymbol{x}; \theta)$ and $g_{\boldsymbol{\mu_x}}(\boldsymbol{z}; \phi)$) and the log variance ($f_{\log \sigma^2_{\boldsymbol{z}}}(\boldsymbol{x}; \theta)$ and $g_{\log \sigma^2_{\boldsymbol{x}}}(\boldsymbol{z}; \phi)$) are computed with neural networks. The prior is considered a centered isotropic multivariate Gaussian $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ of 64 dimensions.

To model the temporal relationship within speech segments, we apply a sequence-to-sequence long short-term memory (Seq2Seq-LSTM) architecture as illustrated in Figure 1. The encoder is a two layer LSTM with 512 hidden units, which inputs the speech segment frame by frame. The outputs from both layers are then concatenated and fed into a fully connected Gaussian parameter layer that predicts the mean and the log variance of the latent variable $\boldsymbol{z}$. The reparameterization trick is applied to rewrite the latent variable as $\boldsymbol{z} = f_{\boldsymbol{\mu_z}}(\boldsymbol{x}; \theta) + \sqrt{\exp(f_{\log \sigma^2_{\boldsymbol{z}}}(\boldsymbol{x}; \theta))} \odot \boldsymbol{\epsilon}$, where $\odot$ denotes the element-wise product, and vector $\boldsymbol{\epsilon}$ is sampled from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

The decoder is also a two layer LSTM with 512 hidden units that takes the sampled latent variable as the input and generates a sequence of outputs. Each output is used as the input to another fully-connected Gaussian parameter layer that predicts the mean and the log variance for one frame of $\boldsymbol{x}$. The entire model can be seen as a stochastic sequence-to-sequence autoencoder that encodes a frame sequence stochastically to the latent space, and then decodes statistically from a sampled latent variable to a sequence of frames.

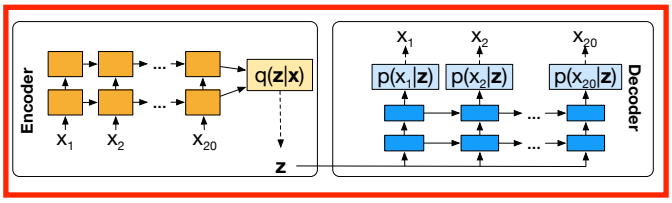


Fig. 1. Illustration of Seq2Seq LSTM VAE architecture.

2.3. Latent Attribute Representations

To losslessly reconstruct speech segments from their latent variables, the latent variables must encode the factors that result in the variability of speech segments. It is proposed and empirically verified in [24] that VAEs learn to use orthogonal subspaces to encode speaker and phone attributes, and the prior distribution of $\boldsymbol{z}$ conditioned on some label r of some type of attribute a is normally distributed. Suppose y_a is the associated label of the attribute a for data $\boldsymbol{x}$, these assumptions can be formulated as follows: $p_\theta(\boldsymbol{z}|y_a = r) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$, and $\boldsymbol{\mu}_{r_i} \perp \boldsymbol{\mu}_{r_j}$ if r_i and r_j are labels of different types of attributes, such as a speaker and a phone.

The mean of the conditional prior $\boldsymbol{\mu}_r$ is defined as the *latent attribute representation*, and can be estimated by averaging the latent variables of speech segments which have the label r. This can be formulated as follows:

$$\boldsymbol{\mu}_r \approx \sum_{i=1}^{N} f_{\boldsymbol{\mu_z}}(\boldsymbol{x}^{(i)}; \theta) \mathbb{1}_{y_a^{(i)}=r} / \sum_{i=1}^{N} \mathbb{1}_{y_a^{(i)}=r}. \quad (2)$$

3. DATA AUGMENTATION METHODS

Here we introduce the idea of nuisance attribute representations in the scenario of speech recognition, and discuss how to compute these representations without supervision. We then summarize how we generate transformed labeled training utterances based on these nuisance attribute representations.

3.1. Nuisance Attributes and VAE-Based Augmentation Method

We define the *nuisance attribute* to be the factors that affect the surface form of a speech utterance but not the linguistic content, such as speaker identity, channel, and background noise, etc. These attributes, unlike phonetic attributes, are generally consistent within an utterance, which implies that we can assume the labels for these attributes are the same for all the segments within an utterance. Therefore, we can compute one *latent nuisance representation* for each utterance using Eq.2. Suppose $\{\boldsymbol{x}^{(i)}_{utt_j}\}_{i=1}^{N_j}$ is the set of segments from an utterance utt_j, we then have $\boldsymbol{\mu}_{utt_j} = \sum_{i=1}^{N_j} f_{\boldsymbol{\mu_z}}(\boldsymbol{x}^{(i)}_{utt_j}; \theta)/N_j$.

We generate labeled training data (i.e. with transcripts) for automatic speech recognition systems by transforming nuisance attributes of the labeled source data. The newly generated data can still use the original transcript for training but differ in some aspect, such as speaker quality and background noise, from the original speech. Figure 2 shows the flowchart of generating transformed labeled data. Let $(\{\boldsymbol{x}^{(i)}_{utt_j}\}_{i=1}^{N_j}, tra_{utt_j})$ be the source utterance of N_j segments with the transcript tra_{utt_j} that we want to modify from. We first encode and sample each segment $\boldsymbol{x}^{(i)}_{utt_j}$ to generate $\boldsymbol{z}^{(i)}_{utt_j}$ using a trained VAE encoder. Then the same modification operation is applied to each latent variable in $\{\boldsymbol{z}^{(i)}_{utt_j}\}_{i=1}^{N_j}$ to produce $\{\tilde{\boldsymbol{z}}^{(i)}_{utt_j}\}_{i=1}^{N_j}$. Finally, we decode $\{\tilde{\boldsymbol{z}}^{(i)}_{utt_j}\}_{i=1}^{N_j}$ using the same trained VAE decoder to obtain the modified utterance $\{\tilde{\boldsymbol{x}}^{(i)}_{utt_j}\}_{i=1}^{N_j}$ that shares the same transcript tra_{utt_j} with the original utterance. In other words, we create new labeled training data: $(\{\tilde{\boldsymbol{x}}^{(i)}_{utt_j}\}_{i=1}^{N_j}, tra_{utt_j})$. We next introduce two types of modification operations in Section 3.2 and 3.3 respectively.

3.2. Type I: Nuisance Attribute Replacement

The first type of modification operation we consider is to replace the nuisance attribute of one utterance with that of another utterance. We assume VAEs use orthogonal subspaces to model phone attributes and nuisance attributes, and apply the operation derived in [24]. Let $\{\boldsymbol{z}^{(i)}_{utt_{src}}\}_{i=1}^{N_j}$ be the encoded latent variables of the source utterance

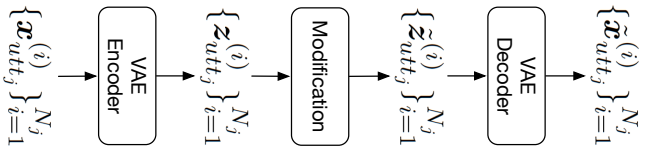
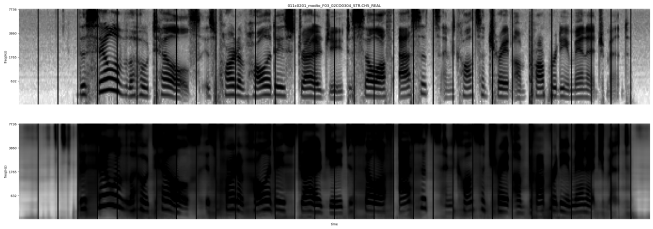


Fig. 2. Flowchart of generating transformed labeled data.

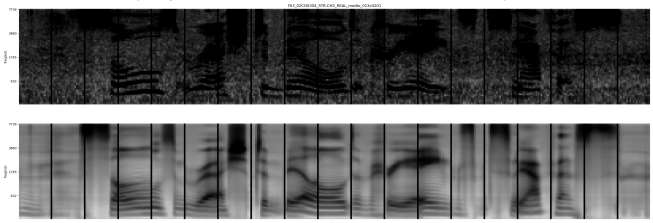
segments we want to modify from, $\boldsymbol{\mu}_{utt_{src}}$ be the latent nuisance representation from the source utterance, and $\boldsymbol{\mu}_{utt_{tar}}$ be the latent nuisance representation from the target utterance. Then we modify $\boldsymbol{z}_{utt_{src}}^{(i)}$ as follows:

$$\tilde{\boldsymbol{z}}_{utt_{src}}^{(i)} = \boldsymbol{z}_{utt_{src}}^{(i)} - \boldsymbol{\mu}_{utt_{src}} + \boldsymbol{\mu}_{utt_{tar}}.$$

Figure 3 shows two examples of modifying the nuisance attributes, where the first row is the original utterance and the second row is the modified utterance. In Figure 3(a), a clean utterance is modified by replacing its nuisance attributes with those from a noisy utterance. Conversely, Figure 3(b) illustrates an example of modifying a noisy utterance by replacing its nuisance attributes with those estimated from a clean utterance. In the figure, segments within an utterance are separated by vertical black lines. From both examples we can observe that while the spacing between harmonics and the level of noise changes, the linguistic content does not seem to change after replacing the nuisance attributes.



(a) modifying a clean utterance to be like a noisy utterance



(b) modifying a noisy utterance to be like a clean utterance

Fig. 3. Two examples of replacing the nuisance attributes.

3.3. Type II: Latent Nuisance Subspace Perturbation

The fundamental assumption of this work is that VAEs learn to use orthogonal subspaces to model linguistic factors and nuisance factors respectively. Hence, we are able to modify the nuisance attribute without changing the original linguistic attribute by only modifying factors in the latent nuisance subspace, but keeping factors in the latent linguistic subspace intact. While the operation in Section 3.2 bypasses the search for the latent nuisance subspace, we can alternatively discover this subspace, and then sample or perturb the factors in it to change the nuisance attribute of an utterance.

3.3.1. Determining the latent nuisance subspace with PCA

Given a dataset of M utterances, we can compute M latent nuisance representations $\{\boldsymbol{\mu}_{utt_j}\}_{j=1}^{M}$, with one for each utterance. The latent nuisance subspace is composed of a set of bases, which captures the variations among these latent nuisance representations. We apply principle component analysis (PCA) on the M latent nuisance representations to obtain a list of eigenvectors $\{\boldsymbol{e}_d\}_{d=1}^{D}$, sorted in a descending order by their associated eigenvalues $\{\sigma_d^2\}_{d=1}^{D}$, where D is the dimension of the latent variable $\boldsymbol{z}$. Each eigenvalue interprets the variance of latent nuisance representations along the direction of its associated eigenvector. We plot the eigenvalues in Figure 4 in a descending order, where we can observe that most of the variation is captured by the first few dimensions.

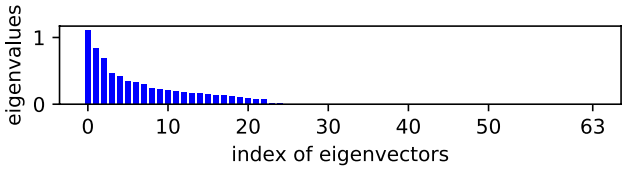


Fig. 4. Eigenvalues of PCA analysis on latent nuisance representations in a descending order.

3.3.2. Soft latent nuisance subspace perturbation

An intuitive way to determine and perturb the latent nuisance subspace is to select the first few eigenvectors and only perturb in those directions. We refer to this as *hard latent nuisance subspace perturbation*, since it demands a hard decision on the rank of the subspace. Alternatively, we propose an approach called *soft latent nuisance subspace perturbation*, which generates a *perturbation vector* $\boldsymbol{p}$ as follows:

$$\boldsymbol{p} = \gamma \sum_{d=1}^{D} \psi_d \sigma_d \boldsymbol{e}_d, \quad \psi_d \sim \mathcal{N}(0,1),$$

where ψ_d is drawn from a normal distribution, σ_d and $\boldsymbol{e}_d$ are square root of d-th largest eigenvalue and its associated eigenvector, and γ is a hyper-parameter, referred to as the *perturbation ratio*. It can be observed that the expected scale we perturb along an eigenvector $\boldsymbol{e}_d$ is proportional to the standard deviation of latent nuisance representations along that eigenvector, which is the square root of its eigenvalue σ_d^2. This approach thus automatically adapts to different distributions of eigenvalues, regardless how many dimensions a VAE learns to use to model the nuisance attributes.

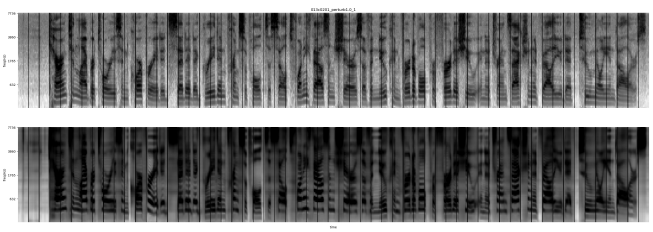


Fig. 5. An example of perturbing latent nuisance attributes.

Let $\{\boldsymbol{z}_{utt_{src}}^{(i)}\}_{i=1}^{N_j}$ be the encoded latent variables of the source utterance segments we want to perturb. We modify each latent vari-

able as follows:

$$\tilde{\boldsymbol{z}}^{(i)}_{utt_{src}} = \boldsymbol{z}^{(i)}_{utt_{src}} + \boldsymbol{p},$$

which adds the same perturbation vector $\boldsymbol{p}$ to each segment in an utterance such that the nuisance attribute change is consistent for all segments within an utterance. Figure 5 shows an example of perturbing the latent nuisance attributes with $\gamma = 1.0$, where the first row is the original utterance and the second row is the perturbed utterance.

4. RELATED WORK

Deep domain adaptation (DDA) to acoustic modeling [19] is a state-of-the-art approach that also addresses the unsupervised domain adaptation problem for robust speech recognition. This work adopts a domain-adversarial training method for neural networks [25], which encourages the networks to learn features that are discriminative for the main learning task, but not discriminative with respect to domains. Specifically, the neural network acoustic model in [19] is composed of one common feature extractor network, and two predictor networks that predict the senone labels and domain labels, respectively. The feature extractor network and the senone predictor network are trained to minimize the senone prediction error and maximize the domain prediction error, while the domain predictor network is trained to minimize the domain prediction error. Hence, during training, unlabeled target-domain data are used to update the feature extractor and domain predictor network, and labeled source-domain data are used to update all three networks.

Our proposed data augmentation method takes a different view by generating more domain-diverse data in order to train more robust models. The two views are complementary, and could be potentially applied in combination.

5. SETUP

5.1. Dataset

Our dataset is based on the CHiME-4 challenge [21], which targets distant-talking ASR and whose setup is based on the speaker-independent medium (5K) vocabulary subset of the Wall Street Journal (WSJ0) corpus [22]. The training set of the CHiME-4 dataset consists of 1,600 utterances recorded in four noisy environments from four speakers, and 7,138 simulated noisy utterances based on the clean utterances in the WSJ0 SI-84 training set. We use the original 7,138 clean utterances as the labeled source-domain data, and the 1600 single channel real noisy utterances as the unlabeled target-domain data for unsupervised domain adaptation. Performance is evaluated on both the real noisy utterances and the original clean utterances in the development partition of the CHiME-4 dataset in terms of the word error rate (WER).

In addition, we also repeat our experiments on Aurora-4 [26] to compare with the results reported in [19]. Aurora-4 is a broadband corpus designed for noisy speech recognition tasks based on WSJ0 as well. Two microphone types, clean/channel are included, and six noise types are artificially added to both microphone types, which results in four conditions: clean(A), channel(B), noisy(C), and channel+noisy(D). We use the clean training set as the labeled source-domain data, and the multi-condition development set as the unlabeled target-domain data. The multi-condition test_eval92 set is used for evaluation.

5.2. VAE Setup and Training

All the original clean utterances and the real noisy utterances are mixed and split into training and development sets with the ratio of 90-10 for training the Seq2Seq LSTM VAE. The VAE is trained with stochastic gradient descent using a mini-batch size of 128 without clipping to minimize the negative variational lower bound plus an $L2$-regularization with weight 10^{-4}. The Adam [27] optimizer is used with $\beta_1 = 0.95$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and initial learning rate of 10^{-3}. Training is terminated if the lower bound on the development set does not improve for 50 epochs.

5.3. ASR Setup and Training

Kaldi [28] is used for feature extraction, decoding, forced alignment, and training of an initial HMM-GMM model on the original clean utterances. The recipe provided by the CHiME-4 challenge (`run_gmm.sh`) and the Kaldi Aurora-4 recipe are adapted by only changing the training data being used. The Computational Network Toolkit (CNTK) [29] is used for neural network-based acoustic model training. For all CHiME-4 experiments, the same LSTM acoustic model [1] with the architecture proposed in [30] is applied, which has 1,024 memory cells and a 512-node projection layer for each LSTM layer, and 3 LSTM layers in total. Following the training setup in [31], LSTM acoustic models are trained with a cross-entropy criterion, using truncated backpropagation-through-time (BPTT) [32] to optimize. Each BPTT segment contains 20 frames, and each mini-batch contains 80 utterances, since we find empirically paralleling 80 utterances has similar performance to 40 utterances.

For all Aurora-4 experiments, the same 6 layer fully-connected deep neural network (DNN) acoustic model with 2,048 hidden units at each layer is applied, which is same architecture as the one in [19], except that the number of hidden units are doubled. The input to the DNN is a context window of 11 frames, with five frames of left context and five frames of right context. Each frame is represented using filter bank features with delta and delta-delta coefficients as proposed in [19]. DNN acoustic models are trained with the cross-entropy criterion, with a mini-batch size of 256. For both LSTM and DNN training, a momentum of 0.9 is used starting from the second epoch [33]. Ten percent of the training data is held out as a validation set to control the learning rate. The learning rate is halved when no gain is observed after an epoch.

We assume for both nuisance attribute replacement and latent nuisance subspace perturbation, the time alignment of senones does not change. Therefore, the same forced alignment is used to train the acoustic models.

6. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we verify the effectiveness of the proposed VAE-based data augmentation methods for unsupervised domain adaptation. On each dataset, the same acoustic model architectures and training procedures, as well as the same language models are used for all the experiments. For the CHiME-4 dataset, besides reporting the WER on the clean and the noisy development sets respectively, we also show the WER for the noisy set by the four recording locations: bus (BUS), cafe (CAF), pedestrian area (PED), and street junction (STR). All the CHiME-4 results are listed in Table 1. For the Aurora-4 dataset, we report the averaged WER as well as the WER in four conditions in Table 2. Different sets of experiments are separated by double horizontal lines and indexed by the *Exp. Index*

Setting			WER (%)		WER (%) in Noisy Condition by Type			
Exp. Index	Aug. Method	Fold	Clean	Noisy	BUS	CAF	PED	STR
1	Orig.	1	**19.04**	**87.80**	96.16	92.35	78.46	84.24
	Recon.	1	19.61	90.72	98.95	93.45	81.52	88.97
2	Repl. Clean	1	**20.03**	67.12	71.99	76.84	55.32	64.33
	Repl. Noisy	1	26.31	**57.66**	62.12	69.25	46.89	52.38
3	Pert., $\gamma = 1.0$	1	20.01	**53.06**	55.66	66.12	41.94	48.50
	Uni-Pert., $\gamma = 1.0$	1	**19.70**	65.07	69.27	75.28	53.65	62.06
	Rev-Pert., $\gamma = 1.0$	1	19.75	87.98	95.13	90.58	76.71	89.50
4	Pert., $\gamma = 0.5$	1	**19.55**	65.61	67.87	77.37	54.54	62.66
	Pert., $\gamma = 1.0$	1	20.01	**53.06**	55.66	66.12	41.94	48.50
	Pert., $\gamma = 1.5$	1	19.99	53.59	57.09	64.91	42.23	50.11
	Pert., $\gamma = 2.0$	1	20.39	58.10	64.35	69.12	45.39	53.55
5	Orig. + Repl. Noisy	2	19.88	55.72	60.72	66.46	45.08	50.63
	Repl. Noisy	2	25.26	55.59	59.24	67.85	44.65	50.63
	Pert., $\gamma = 1.0$	2	**19.82**	**52.49**	55.52	65.04	41.17	48.24

Table 1. CHiME-4 development set word error rate of acoustic models trained on different augmented sets.

on the first column. The second column, *Aug. Method*, explains the augmentation method and the hyper-parameter being used. The ratio of the new training set to the original clean training set is listed on the third column, referred to as the *Fold*.

6.1. Baselines

We first establish baselines by training models on two sets. The first set, *Orig.*, refers to the original clean training set that does not involve VAE. The second set, *Recon.*, refers to the reconstructed clean training set that is generated by using the VAE to first encode and then decode. Note that this does not involve the modification methods mentioned in Sections 3.2 and 3.3.

The results are listed in Table 1, *Exp. Index 1*. The fourth row shows the results on the matched domain (clean), and the fifth row shows the results on the mismatched domain (noisy). It can be observed that the performance degrades significantly when the models are tested on the mismatched domain. The WER increases from 19.04% to 87.08% for *Orig.*, and from 19.61% to 90.72% for *Recon.* respectively. In addition, since the reconstruction from the VAE is not perfect, part of the information may be lost during this process. Hence, the model trained on *Recon.* is slightly worse than the one trained on *Orig.* for all testing conditions. Lastly, the relative WERs of the four location are consistent on the both training sets. BUS appears to be the most difficult one, while PED is the easiest one among the four locations.

6.2. Replacing Nuisance Attributes

We evaluate the effectiveness of augmenting data by replacing the nuisance attributes as mentioned in Section 3.2. Let $\mathcal{U}_{src} = \{\boldsymbol{\mu}_{utt_j}\}_{j=1}^{M_{src}}$ be the set of latent nuisance representations of the source domain utterances, and $\mathcal{U}_{tar} = \{\boldsymbol{\mu}_{utt_j}\}_{j=M_{src}}^{M}$ be the set of latent nuisance representations of the target domain utterances. M_{src} is the number of source domain utterances, and $M_{tar} = M - M_{src}$ is the number of target domain utterances. We create the augmented set *Repl. Clean* by replacing the latent nuisance representation of each source domain utterance with one drawn from $\mathcal{U}_{src}$. The *Repl. Noisy* is generated similarly but is replaced with one drawn from $\mathcal{U}_{tar}$.

The results are shown in Table 1, *Exp. Index 2*. For both augmented methods, we observe at least 20% absolute WER reduction on the target domain compared to the baselines. We observe an additional 10% absolute WER reduction when replacing the latent nuisance representations with those taken from the target domain instead of the source domain. We also observe that *Repl. Noisy* shows 6% worse WER on the source domain than *Repl. Clean*. The relative strength of *Repl. Clean* and *Repl. Noisy* on different domains verifies the effectiveness of our proposed method at shifting the distribution from one domain to another.

6.3. Correctness of Soft Latent Nuisance Subspace Perturbation

We first examine the correctness of our proposed soft latent nuisance subspace perturbation by proposing two alternative perturbation methods. To eliminate the effect of the perturbation scale on the performance, we consider two alternative methods subject to the constraint that the expected squared Euclidean norm of the perturbation vector $\boldsymbol{p}$ is the same as the proposed method.

Recall that $\boldsymbol{p} = \gamma \sum_{d=1}^{D} \psi_d \sigma_d \boldsymbol{e}_d$ for our proposed method. We then have: $\mathbb{E}\left[||\boldsymbol{p}||_2^2\right] = \gamma^2 \sum_{d=1}^{D} \sigma_d^2 \mathbb{E}\left[\psi_d^2\right] = \gamma^2 \sum_{d=1}^{D} \sigma_d^2$. We then consider *uniform perturbation* to be a method that perturbs each direction with the same expected scale, controlled by the same perturbation ratio hyper-parameter γ. The perturbation vector $\boldsymbol{p}_{uni}$ derived from this method can be formulated as $\boldsymbol{p}_{uni} = \gamma \sum_{d=1}^{D} \psi_d \sigma_{uni} \boldsymbol{e}_d$, where $\sigma_{uni} = \sqrt{\sum_{d=1}^{D} \sigma_d^2 / D}$. In addition, we design another method that reverses the expected perturbation scales from the proposed soft latent nuisance subspace perturbation method, named the *reverse soft latent nuisance subspace perturbing*. The perturbation vector $\boldsymbol{p}_{rev}$ can be formulated as $\boldsymbol{p}_{rev} = \gamma \sum_{d=1}^{D} \psi_d \sigma_{D-d} \boldsymbol{e}_d$.

We show the results of soft latent nuisance subspace perturbation (Pert.), uniform perturbation (Uni-Pert.), and reverse soft latent nuisance subspace perturbation (Rev-Pert.) in Table 1, *Exp. Index 3*, which all use the same perturbation ratio $\gamma = 1.0$. We can clearly observe the superiority of the proposed method among the three methods applied on the target domain. Pert. reduces the absolute WER by almost 35% from the baseline and outperforms Uni-Pert. by 12%, while Rev-Pert. achieves almost no improvement. This experiment verifies the importance of determining an appropriate way to perturb the latent space and the correctness of our method.

Setting			WER (%)	WER (%) by Condition			
Exp. Index	Aug. Method/Baselines	Fold	Avg.	Clean(A)	Noisy(B)	Channel(C)	Channel+Noisy(D)
0	Clean-DNN-HMM [19]	-	36.22	3.36	29.74	21.02	50.73
	DDA-DNN-HMM [19]	-	22.53	3.24	14.52	17.82	34.55
	DNN-PP [34]	-	18.7	5.1	12.0	10.5	29.0
1	Orig.	1	53.98	3.38	50.56	42.67	67.70
	Recon.	1	66.29	4.58	65.44	51.02	79.97
2	Repl. Noisy	1	22.53	4.80	16.31	14.72	32.99
3	Pert., $\gamma = 0.5$	1	35.37	4.11	27.73	33.51	48.52
	Pert., $\gamma = 1.0$	1	24.82	4.35	17.11	22.38	36.36
	Pert., $\gamma = 1.5$	1	21.98	4.24	15.08	16.87	32.69
	Pert., $\gamma = 2.0$	1	20.68	4.45	14.33	14.74	30.72
	Pert., $\gamma = 2.5$	1	20.99	4.99	15.35	15.54	30.22
	Pert., $\gamma = 3.0$	1	21.18	5.29	15.47	15.71	30.45
	Pert., $\gamma = 3.5$	1	21.33	5.45	16.13	14.70	30.29
	Pert., $\gamma = 4.0$	1	22.00	6.43	17.15	15.00	30.62
4	Pert., $\gamma = 2.0$	2	20.06	4.13	13.85	14.96	29.77
	Pert., $\gamma = 2.0$	4	19.42	4.09	13.34	14.14	28.92
	Pert., $\gamma = 2.0$	8	18.86	4.28	12.89	13.51	28.16
	Pert., $\gamma = 2.0$	16	**18.76**	4.04	12.84	13.54	28.01

Table 2. Aurora-4 test_eval92 set word error rate of acoustic models trained on different augmented sets.

6.4. Effect of Perturbation Ratios

We next examine the effect on the hyper-parameter γ by choosing four scales: 0.5, 1.0, 1.5, 2.0, and list the results in Table 1, *Exp. Index 4*. We observe different WER trends for different perturbation ratios for the two testing conditions. First, regarding the target domain WER, we notice that $\gamma = 1.0$ reaches the best performance among the four scales. The smaller the perturbation ratio is, the more similar to the original clean data the augmented perturbed data would be. Hence, when we decrease the perturbation ratio, the performance would asymptotically approach those of the original clean data. On the other hand, as we increase the perturbation ratio, the chance of the perturbed utterances becoming linguistically different increases. This may hurt the performance and cancel out the benefit of having more diverse data by perturbing the nuisance attributes. As for the source domain WER, we observe degradation when increasing the perturbation ratio, because the perturbed data distribution becomes less similar to original clean data distribution.

6.5. Effect of Dataset Size

In this section, we study the effect of the size by combining different sets of augmented data or the original data. Specifically, three cases are considered: (1) combining *Repl. Noisy* with the original data. (2) combining *Repl. Noisy* with another copy of *Repl. Noisy*. (3) combining *Pert.,* $\gamma = 1.0$ with another copy of *Pert.,* $\gamma = 1.0$.

The results are listed in Table 1, *Exp. Index 5*. In the first two cases, both source and target domain WERs are improved from the one-fold *Repl. Noisy*. While adding another copy of *Repl. Noisy* shows slightly better (0.13%) WER in the target domain of the first two cases, adding the original clean data significantly reduces (6.43%) WER in the source domain. This suggests that the second case addresses the issue of *Repl. Noisy* on shifting the data distribution entirely to the target domain. In the third case, a slight but consistent 0.19% and 0.57% WER reductions from the one-fold *Pert.,* $\gamma = 1.0$ in the source and target domain are observed. In summary, all three cases show improvement by increasing the size.

6.6. Comparing with DDA on Aurora-4

In this section, we repeat the experiments on the Aurora-4 dataset and compare with deep domain adaptation. Table 2 listed our Aurora-4 results and the reference results. DNN-PP [34] is a method compared in [19] that requires parallel clean-noisy data for training an speech enhancement model as a preprocessor.

Baseline results are established in Table 2, *Exp. Index 1*, where the models are trained with the original clean features (*Orig.*), and the VAE-reconstructed clean features (*Recon.*), respectively. Here we can observe significant degradation on mismatched domains (B, C, and D) from the matched domain (A). Results of nuisance attribute replacement and soft latent nuisance subspace perturbation with different perturbation ratios are shown in Table 2, *Exp. Index 2* and *Exp. Index 3*. Both augmentation methods achieve roughly 30% absolute WER reduction, and the soft latent nuisance subspace perturbation reaches the best performance when using a perturbation ratio $\gamma = 2.0$. By increasing the dataset size, we observe further WER reduction from two-fold to 16-fold.

Since the detailed training recipe is not provided in [19], we could not reproduce exactly the same baseline results. However, despite the fact that our baseline is 17.76% worse than that in [19], our best system (18.76%) still achieves better performance than the result of DDA (22.53%) that was reported in [19], and matches the results of DNN-PP [34] without the need for parallel data.

7. CONCLUSION AND FUTURE WORK

In this paper, we present two VAE-based data augmentation methods for unsupervised domain adaptation to robust ASR. In particular, we study the latent representations obtained from VAEs, which enable us to transform nuisance attributes of speech through modifying the latent variables. Our proposed methods are evaluated two datasets, and achieve about 35% absolute WER reduction on both sets. For future work, we plan to investigate the proposed augmentation method using more advanced FHVAE models [35], which explicitly disentangle sequence and segment level attributes in the latent space.

8. REFERENCES

[1] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *Interspeech*, 2014, pp. 338–342.

[2] Tara N Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdel-rahman Mohamed, George Dahl, and Bhuvana Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[3] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts.," in *INTERSPEECH*, 2015, pp. 3214–3218.

[4] Wei-Ning Hsu, Yu Zhang, and James Glass, "A prioritized grid long short-term memory rnn for speech recognition," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 467–473.

[5] Haşim Sak, Félix de Chaumont Quitry, Tara Sainath, Kanishka Rao, et al., "Acoustic modelling with cd-ctc-smbr lstm rnns," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 604–609.

[6] Xavier Anguera, Chuck Wooters, and Javier Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[7] Hakan Erdogan, Tomoki Hayashi, John R Hershey, Takaaki Hori, Chiori Hori, Wei-Ning Hsu, Suyoun Kim, Jonathan Le Roux, Zhong Meng, and Shinji Watanabe, "Multi-channel speech recognition: Lstms all the way through," in *CHiME-4 workshop*, 2016.

[8] Arun Narayanan and DeLiang Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7092–7096.

[9] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[10] Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng, "Recurrent neural networks for noise reduction in robust asr," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[11] Xue Feng, Yaodong Zhang, and James Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1759–1763.

[12] Brian ED Kingsbury, Nelson Morgan, and Steven Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech communication*, vol. 25, no. 1, pp. 117–132, 1998.

[13] Richard M Stern and Nelson Morgan, "Features based on auditory physiology and perception," *Techniques for Noise Robustness in Automatic Speech Recognition*, p. 193227, 2012.

[14] Oriol Vinyals and Suman V Ravuri, "Comparing multilayer perceptron to deep belief network tandem features for robust asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4596–4599.

[15] Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4153–4156.

[16] Mark JF Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[17] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[18] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[19] Sining Sun, Binbin Zhang, Lei Xie, and Yanning Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, 2017.

[20] Jinyu Li, Dong Yu, Jui-Ting Huang, and Yifan Gong, "Improving wideband speech recognition using mixed-bandwidth training data in cd-dnn-hmm," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 131–136.

[21] Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.

[22] John Garofalo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[23] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," in *Interspeech*, 2017, pp. 1273–1277.

[25] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[26] David Pearce, *Aurora working group: DSR front end LVCSR evaluation AU/384/02*, Ph.D. thesis, Mississippi State University, 2002.

[27] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[29] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep., Tech. Rep. MSR, Microsoft Research, 2014, http://codebox/cntk, 2014.

[30] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass, "Highway long short-term memory RNNs for distant speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5755–5759.

[31] Wei-Ning Hsu, Yu Zhang, Ann Lee, and James R Glass, "Exploiting depth and highway connections in convolutional recurrent deep neural networks for speech recognition.," in *INTERSPEECH*, 2016, pp. 395–399.

[32] Ronald J Williams and Jing Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.

[33] Yu Zhang, Dong Yu, Michael L Seltzer, and Jasha Droppo, "Speech recognition with prediction-adaptation-correction recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5004–5008.

[34] Jun Du, Qing Wang, Tian Gao, Yong Xu, Li-Rong Dai, and Chin-Hui Lee, "Robust speech recognition with speech enhanced deep neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[35] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in Neural Information Processing Systems*, 2017.